Topological Quantum Computing and the Jones Polynomial

Samuel J. Lomonaco, Jr.[a] and Louis H. Kauffman[b]

[a]Department of Computer Science and Electrical Engineering, University of Maryland
Baltimore County (UMBC), Baltimore, Maryland 21250, USA
[b]Department of Mathematics, Statistics, and Computer Science, University of Illinois at
Chicago (UIC), Chicago, Illinois 60607, USA

ABSTRACT

In this paper, we give a description of a recent quantum algorithm created by Aharonov, Jones, and Landau for approximating the values of the Jones polynomial at roots of unity of the form $e^{2\pi i/k}$. This description is given with two objectives in mind. The first is to describe the algorithm in such a way as to make explicit the underlying and inherent control structure. The second is to make this algorithm accessible to a larger audience.

Contents

Send correspondence to S.J.L: E-mail: Lomonaco@umbc.edu

1. INTRODUCTION

In this paper, we give a description of a recent quantum algorithm created by Aharonov, Jones, and Landau[1] for approximating the values of the Jones polynomial at roots of unity of the form $e^{2\pi i/k}$, for k a positive integer. We do so with two objectives in mind. The first is to describe the algorithm in such a way as to make explicit the underlying and inherent control structure. The second is to make this algorithm accessible to a larger audience.

To avoid cluttering this exposition, we focus solely on the version of this quantum algorithm based on the Markov trace closure of a braid. The alternative platt closure version is left as an exercise for the reader.

Readers already familiar with knot theory may want to skip to section 7.

2. KNOT THEORY IS ...

In its most general form, knot theory is the study of the fundamental problem of placement:

The Placement Problem. *When are two placements of a space X in a space Y the same or different?*
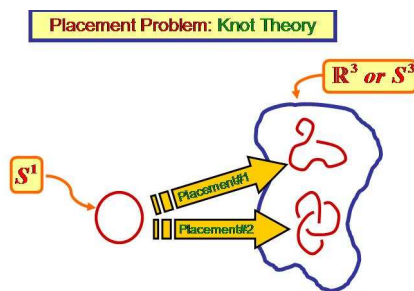

Figure 1. The Placement Problem.

In its most renowned form, knot theory is the study of the placement of a 1-sphere* S^1(or a disjoint union of 1-spheres) in real 3-space $\mathbb{R}^3$ (or the 3-sphere S^3), called the **ambient space**. In this case, "placement" usually means a smooth (or piecewise linear) embedding, i.e., a smooth homeomorphism into the ambient space. Such a placement is called a **knot** if a single 1-sphere is embedded (or a **link**, if a disjoint union of many 1-spheres is embedded.)

Two knots (or links) are said to be the same, i.e., of the **same knot type**, if there exists an orientation preserving autohomeomorphism† of the ambient space carrying one knot into the other. Otherwise, they are said to be different, i.e., of **different knot type**. Such knots are frequently represented by a knot diagram, i.e., a planar 4-valent graph with vertices appropriately labelled as undercrossings/overcrossings, as shown in figure 2.

---

*By 1-sphere we mean a circle.
†A provenly equivalent definition is that two knots are of the same knot type if and only if there exists an isotopy of the ambient space that carries one knot onto the other.
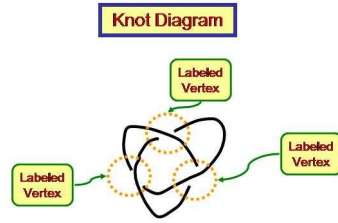
Figure 2. A knot diagram.

The fundamental problem of knot theory can now be stated as:

The Fundamental Problem of Knot Theory. *When are two knots of the same or of different knot type?*

A useful knot theoretic research tool is Reidemeister's theorem, which makes use of the Reidemeister moves as defined in figure 3.:

THEOREM 2.1 (**Reidemeister**). *Two knot diagrams represent the same knot type if and only if it is possible to transform one into the other by applying a finite sequence of Reidemeister moves.*
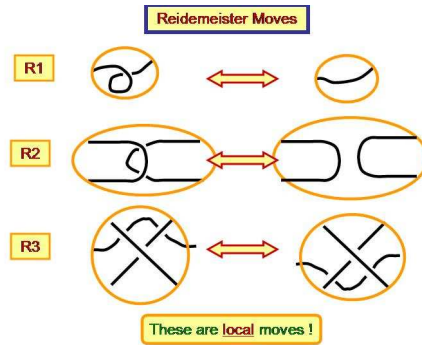

Figure 3. The Reidemeister moves.

The standard approach to attacking the fundamental problem of knot theory is to create knot invariants for distinquishing knots. By a **knot invariant** I we mean a map from knots to a specified mathematical domain which maps knots (or links) of the same type to the same mathematical object. *Thus, if an invariant is found to be different on two knots (or links), then the two knots (or links) cannot be of the same knot type!*

The Jones polynomial is one such invariant, and indeed a very significant one. The Jones polynomial maps each knot (or link) to a Laurent polynomial with integer coefficients, i.e., an invariant with domain the Laurent ring $\mathbb{Z}\left[t, t^{-1}\right]$. If two knots (or links) have different Jones polynomials then they must be different. A definition of this famous knot invariant is given in section 6 of this paper.

But before we define the Jones polynomial we will need to look at a group invented by Emil Artin, i.e., the braid group, which is now beginning to have an impact on quantum computing.

3. THE BRAID GROUP B_N

DEFINITION 3.1. *The n-**stranded braid group** B_n is is the group generated by the symbols $b_1, b_2, \ldots, b_{n-1}$ subject to the defining relations*

$$\begin{cases} b_i b_{i+1} b_i = b_{i+1} b_i b_{i+1} & \text{for } 1 \leq i < n \\ b_i b_j = b_j b_i & \text{for } |i - j| \geq 2 \end{cases}$$

The braid group B_n is easily understood in terms of diagrammatics, as shown in Figures 4 to 13.

As illustrated in Figure 4, an element of the braid group B_3 can be thought of as a hatbox with the three points at the top connected to the three points at the bottom by smooth nonintersecting curves, called **strands**. As illustrated in Figures 5 and 6, two such elements of B_n are equal if it is possible to continuously move within the hatbox the strands of one braid into the strands of the other without cutting or breaking the strands and without letting the strands pass through one another. Figure 5 shows two equal braids, and Figure 6 shows two braids that cannot be equal. As illustrated in Figure 7, the enclosing hatbox is usually omitted, but understood to be there.

The product of two elements of B_n is defined, as illustrated in Figure 8, by simply stacking one hatbox on top of the other. Under this definition of multiplication, it can be shown that every braid has a multiplicative inverse. An example is given in Figure 9 of the inverse of a braid. The braid group actually has a finite set of generators, as shown in Figure 10. A complete set of defining relations among these generators are shown in Figures 11 and 12.
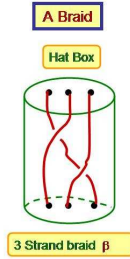

Figure 4. An example of a three stranded braid in B_3.
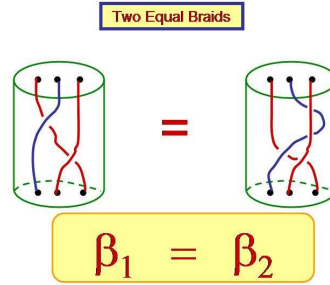

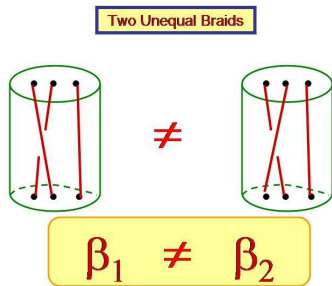**Figure 5. An example of two equal braids.**

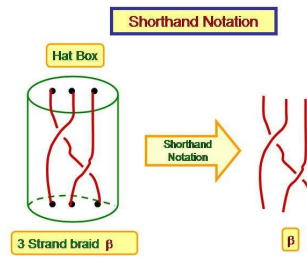**Figure 6.** An example of two non-equal braids.

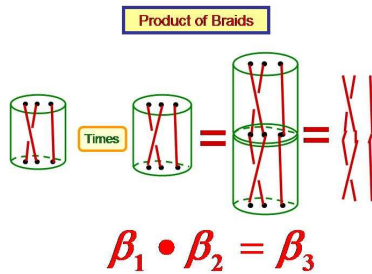**Figure 7.** Shorthand notation for braids.

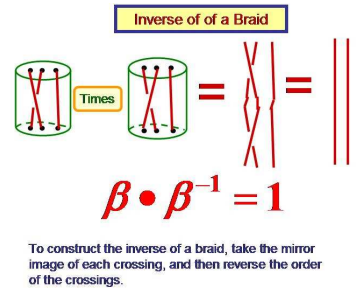
Figure 8. The product of braids.
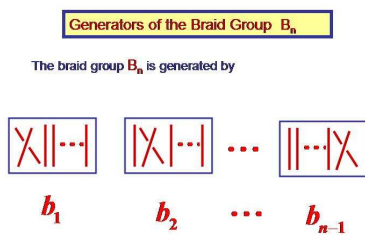

Figure 9. The inverse of a braid.

Figure 10. The generators of the braid group.
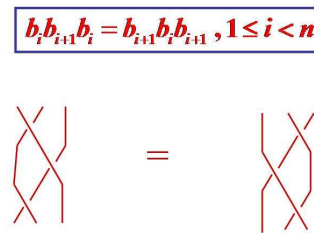
Figure 11. A diagrammatic illustration of one of the first set of defining relations.

$$b_i b_j = b_j b_i \,, \text{ for } |i-j| \geq 2$$

Figure 12. A diagrammatic
illustration of one of the second set
of defining relations.
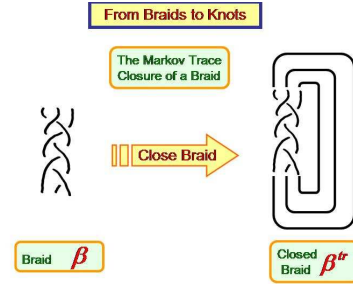
Figure 13. The closure (or trace) of
a braid.

4. TRANSFORMING BRAIDS INTO KNOTS AND VICE VERSA: THE CLOSURE (OR TRACE) OF A BRAID

Why are braids of importance in knot theory?

The answer to this questions begins by observing that every braid β can be converted into a knot (or link) by forming the closure (a.k.a., trace) β^{tr} as shown in Figure 13. Then, of course, there is the famous Markov theorem telling us when two braids produce the same knot (or link):

THEOREM 4.1 (MARKOV). *Two braids β_1 and β_2 produce the same knot (or link) under braid closure if and only if there exists a finite sequence of Markov moves that transforms one braid into the other.*

The two Markov moves M_1 and M_2 are are shown in Figures 14 and 15.
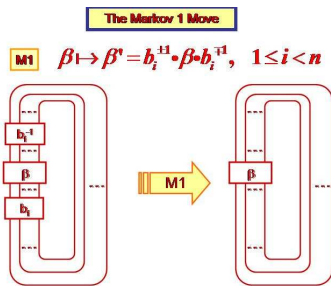
Figure 14. The Markov 1 move.
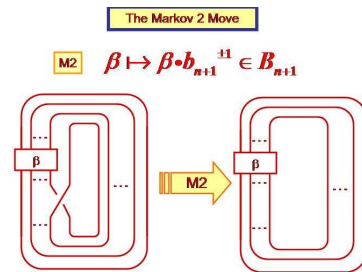
Figure 15. The Markov 2 move.

Most amazingly, the process of transforming a braid into a knot (or link) can be reversed, as stated by Alexander's theorem:

THEOREM 4.2 (ALEXANDER). *Every knot (or link) is the closure of a braid.*

Returning to our original highly algebraic definition of the braid group B_n, we should mention that each braid β in B_n can be expressed as a product of the generators $b_1, b_2, \ldots, b_{n-1}$ and their inverses $b_1^{-1}, b_2^{-1}, \ldots, b_{n-1}^{-1}$. Thus, each braid β can be written in the form

$$\beta = \prod_{i=1}^{\ell} b_{j(i)}^{\epsilon(i)} = b_{j(1)}^{\epsilon(1)} b_{j(2)}^{\epsilon(2)} \cdots b_{j(\ell)}^{\epsilon(\ell)} \ ,$$

where $\epsilon(i) = \pm 1$ for $i = 1, 2, \ldots, \ell$. We will call such a product in the generators $b_1, b_2, \ldots, b_{n-1}$ and their inverses $b_1^{-1}, b_2^{-1}, \ldots, b_{n-1}^{-1}$ a **word** defining the braid β. Two words define the same braid β if and only if it is possible transform one into the other by applying a finite sequence of Tieze transformations. For a definition of Tieze transformations, we refer the reader to Crowell and Fox.[5]

Finally, we define the **writhe** of a braid β, written $Writhe(\beta)$, as the sum of the exponents in any word defining the braid.

5. THE TEMPERLEY-LIEB ALGEBRA $TL_N(D)$

Our next stepping stone to the definition of the Jones polynomial is an algebra, called the Temperley-Lieb algebra.

DEFINITION 5.1. *Let d be an indeterminate complex number. Then for each positive integer n, the **Temperley-Lieb algebra** $TL_n(d)$ is defined as the algebra with identity 1 generated by the identity 1 and the symbols*

$$E_1, E_2, \ldots, E_{n-1}$$

subject to the defining relations

$$\begin{cases} E_i E_j = E_j E_i & for \quad |i - j| \geq 2 \\[2mm] E_i E_{i \pm 1} E_i = E_i & for \quad 1 \leq i < n \\[2mm] E_i^2 = d E_i & for \quad 1 \leq i < n \end{cases}$$

The Temperley-Lieb algebra $TL_n(d)$ is easily understood in terms of diagrammatics[‡], as illustrated in Figures 4 through 13.

As shown in Figure 13, an element of the Temperley-Lieb algebra $TL_3(d)$ can be thought of as the algebra consisting of all linear combinations of rectangles, each rectangle with 3 points at the top and 3 points at the bottom connected by smooth non-intersecting curves, called **strands**. As shown in Figures 17 and 18, two such elements of $TL_n(d)$ are equal if it is possible to continuously move the strands of one into the strands of the other without cutting or breaking the strands and without letting the strands pass through each other. Figure 17 shows two equal elements, and Figure 18 shows two non-equal elements. As shown in Figure 19, the enclosing rectangle is usually omitted, but understood to be there.

The product of two elements of $TL_n(d)$ is defined, as shown in Figure 20, i.e., by stacking one rectangle on top of the other. As illustrated in Figure 21, if a circle should happen to arise as a result of the product, then it is simply replaced by the indeterminate d times the resulting rectangle with the circle omitted. The generators of the Temperley-Lieb algebra are shown in Figure 22. We leave, as an amusing exercise for the reader, the task of translating the complete set of defining relations for the Temperley-Leib algebra given above into diagrammatics.

[‡]The diagrammatic representation of the Temperler-Leib algebra is due to Kauffman.[18,20,21]

Rectangle

An element X of TL$_n$(d)

Figure 16.

Two Equal Elements of TL$_n$(d)

$=$

$X_1 = X_2$

Figure 17.

Two Unequal Elements of TL$_n$(d)

$\neq$

$$X_1 \neq X_2$$

Figure 18.

Shorthand Notation

Rectangle

Shorthand Notation

X   X

Figure 19.

Product

Times $=$ $=$

$$X_1 \bullet X_2 = X_3$$

Figure 20.

Product

Times $=$ $=$ $= d$
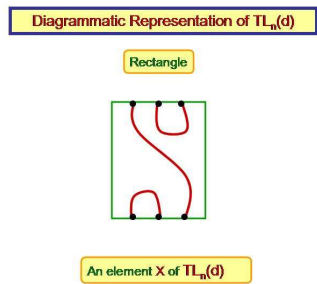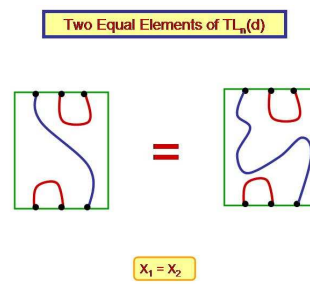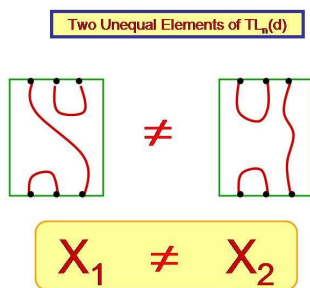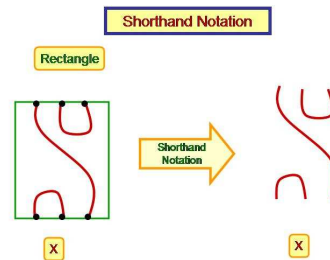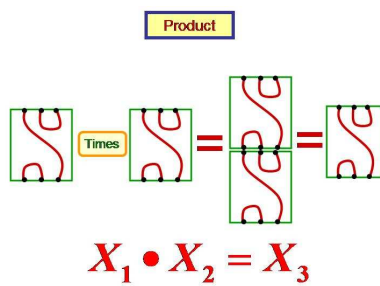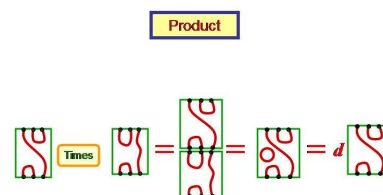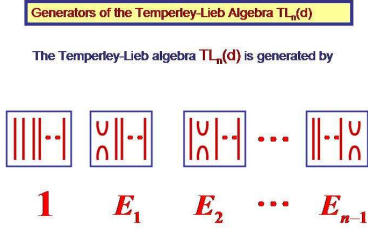
Figure 21.

Figure 22. The Temperley-Lieb algebra $\mathbf{TL}_n(d)$ is generated by all formal sums of products of the above generators.
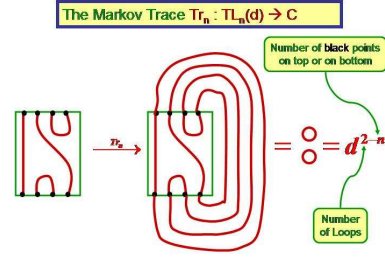
Figure 23. The diagramatic definition of the Markov trace $Tr_n : TL_n(d) \longrightarrow \mathbb{C}$.

We would be amiss if we did not mention that there is a map $Tr_n : TL_n(d) \longrightarrow \mathbb{C}$ from the Temperley-Lieb algebra $TL_n(d)$ to the complex numbers $\mathbb{C}$, called the **Markov trace**, satisfying the following three conditions:

- $Tr_n(1) = 1$

- $Tr_n(XY) = Tr_n(YX)$ for all X and Y in $TL_n(d)$

- If $X \in TL_n(d)$, then $Tr_{n+1}(XE_n) = \frac{1}{d} Tr_n(X)$

A diagrammatic definition of the Markov trace is shown in figure 23.

We will later need the following theorem:

THEOREM 5.2. *The above three conditions uniquely determine the Markov trace, i.e., any map $TL_n(d) \longrightarrow \mathbb{C}$ satisfying the above three conditions must be the same as that defined by figure 23.*

6. THE DEFINITION OF THE JONES POLYNOMIAL

Let d be an indeterminate complex number, and let A also be an indeterminant complex number such that $d = -A^2 - A^{-2}$. Let $TL_n(d)$ be the corresponding Temperley-Lieb algebra, and let B_n denote the n-stranded braid group. Then the **Jones representation**

$$\rho_A : TL_n(d) \longrightarrow B_n$$

is the group representation defined by

$$\begin{cases} b_i & \longmapsto & AE_i + A^{-1}1 \\ \\ b_i^{-1} & \longmapsto & A^{-1}E_i + A1 \end{cases}$$

where $b_1, b_2, \ldots, b_{n-1}$ denote the generators of the braid group B_n, and where $1, E_1, E_2, \ldots, E_{n-1}$ denote the generators of the Temperley-Lieb algebra $TL_n(d)$.

We leave for the reader's amusement the exercise of verifying that the images under ρ_A of the generators $b_1, b_2, \ldots, b_{n-1}$ satisfy the defining relations of the braid group.

We are now finally in a position to define the Jones polynomial.

Let β be an element of the n-stranded braid group B_n, and let β^{Tr} denote the knot (or link) constructed from the closure of the braid β. Then the **Jones polynomial** $V_{\beta^{Tr}}(t)$ of the knot (or link) β^{Tr} is the Laurent polynomial in the polynomial ring $\mathbb{Z}[t, t^{-1}]$ over the integers $\mathbb{Z}$ given by

$$V_{\beta^{Tr}}\left(A^{-4}\right) = -A^{2Writhe(\beta)} d^{n-1} Tr_n\left(\rho_A(\beta)\right) \quad ,$$

where $t = A^{-4}$, where $Writhe\left(\beta^{Tr}\right)$ denotes the writhe[§] of the braid β, and where $Tr_n\left(\rho_A(\beta)\right)$ denotes the Markov trace of the value of the Jones representation ρ_A on the braid β.

7. THE REPRESENTATION $\Phi : TL_N(D) \longrightarrow \mathbb{C}U\left(\mathcal{H}_{N,K}\right)$ OF THE TEMPERLEY-LIEB ALGEBRA $TL_N(D)$

Our objective is to describe the polytime quantum algorithm in AJL[1] for approximating values of the Jones polynomial $V_{\beta^{Tr}}(t)$ at the primitive k-th roots of unity $t = e^{2\pi i/k}$ for positive integers k. To this end, we begin by constructing a representation of the Temperley-Lieb algebra $TL_n(d)$ which carries the image of the Jones representation $\rho_A : B_n \longrightarrow TL_n(d)$ onto a group of unitary transformations.

Let G_k denote the graph of $k-1$ vertices and $k-2$ edges given in Figure 24.
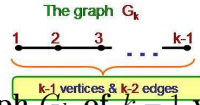
Figure 24. The graph G_k of $k-1$ vertices and $k-2$ edges.

The adjacency matrix M_k of the graph G_k is easily seen to be

$$M_k = \begin{pmatrix} 0 & 1 & 0 & \cdots & & 0 & 0 \\ 1 & 0 & 1 & & & 0 & 0 \\ 0 & 1 & 0 & & & & \\ & \vdots & & & \ddots & & \\ 0 & 0 & 0 & & & 0 & 1 \\ 0 & 0 & 0 & \cdots & & 1 & 0 \end{pmatrix}$$

Moreover, $d = 2\cos(\pi/k)$ can be shown to be an eigenvalue of M_k corresponding to the eigenvector

$$\vec{\lambda} = (\lambda_1, \lambda_2, \ldots, \lambda_{k-1}) = (\ \sin(\pi\ell/k)\ )_{1 \leq \ell < k}$$

We can now construct our representation as follows:

[§]Please refer to section 2 for a definition of writhe.

Let $\mathcal{P}_{n,k}$ denote the set of paths in the graph G_k of **length** n starting at the vertex 1, and let $\mathcal{H}_{n,k}$ denote the Hilbert space with orthonormal basis

$$\{\, |p\rangle : p \in \mathcal{P}_{n,k}\}$$

with basis elements labelled by the paths p in $\mathcal{P}_{n,k}$.

Interpreting 0 as "to the left" and 1 as "to the right," we identify each length n path p with a binary string of length n.

For each path p, let $p^{i-1\rfloor}$ be the subpath corresponding to the first $i-1$ bits of p, let $p^{\lfloor i\cdots i+1\rfloor}$ denote the subpath corresponding to bits i up to and including bit $i+1$ of p, and finally let $p^{\lfloor i+2}$ denote the subpath of p corresponding to bits $i+2$ up to and including the last n-th bit. Let $e_i(p)$ be the endpoint of the subpath $p^{i-1\rfloor}$. Hence, $e_i(p) \in \{1, 2, \ldots, k-1\}$.

Select $d = 2\cos(\pi/k)$. Since A is related to d via the formula $d = -A^2 - A^{-2}$, we choose among the first four possible choices $\pm e^{i\pi/(2k)}$ for A the value

$$A = e^{-i\pi/(2k)}$$

We are now ready to define a representation

$$\Phi : TL_n(d) \longrightarrow \mathbb{C}U(\mathcal{H}_{n,k})$$

of the Temperley-Lieb algebra $TL_n(d)$ into the group ring $\mathbb{C}U(\mathcal{H}_{n,k})$ of the group $U(\mathcal{H}_{n,k})$ of unitary transformations on the Hilbert space $\mathcal{H}_{n,k}$ by specifying the images

$$\Phi_i = \Phi(E_i)$$

of each of the generators E_i of the Temperley-Lieb algebra $TL_n(d)$, taking great care to make sure that the Φ_i's satisfy the same defining relations as the E_i's.

We define Φ_i as:

$$\Phi_i |p\rangle = \begin{cases} 0 & \text{if } p^{\lfloor i\cdots i+1\rfloor} = 00 \\[2mm] \frac{\lambda_{e_i(p)-1}}{\lambda_{e_i(p)}} |p\rangle + \frac{\sqrt{\lambda_{e_i(p)-1}\lambda_{e_i(p)+1}}}{\lambda_{e_i(p)}} |p^{i-1\rfloor}10p^{\lfloor i+2}\rangle & \text{if } p^{\lfloor i\cdots i+1\rfloor} = 01 \\[2mm] \frac{\sqrt{\lambda_{e_i(p)-1}\lambda_{e_i(p)+1}}}{\lambda_{e_i(p)}} |p^{i-1\rfloor}01p^{\lfloor i+2}\rangle + \frac{\lambda_{e_i(p)+1}}{\lambda_{e_i(p)}} |p\rangle & \text{if } p^{\lfloor i\cdots i+1\rfloor} = 10 \\[2mm] 0 & \text{if } p^{\lfloor i\cdots i+1\rfloor} = 11 \end{cases}$$

We leave for the reader the exercise of showing that the transformations Φ_i satisfy the defining identities of the E_i's, i.e., the identities

$$\begin{cases} \Phi_i \Phi_j = \Phi_j \Phi_i & \text{for } |i-j| \geq 2 \\[2mm] \Phi_i \Phi_{i\pm 1} \Phi_i = \Phi_i & \text{for } 1 \leq i < n \\[2mm] \Phi_i^2 = d\Phi_i & \text{for } 1 \leq i < n \end{cases}$$

and hence that Φ is a legitimate representation of the Temperley-Leib algebra $TL_n(d)$.

8. CONSTRUCTING A TRACE $\widetilde{TR}$ COMPATIBLE WITH THE MARKOV TRACE TR_N

We next need to construct a trace $\widetilde{Tr}$ on the image of the representation $\Phi : TL_n(d) \longrightarrow \mathbb{C}U(\mathcal{H}_{n,k})$ which is compatible with the Markov trace Tr_n, i.e., a trace $\widetilde{Tr}$ such that the following diagram is commutative

$$
\begin{array}{ccc}
TL_n(d) & \xrightarrow{\Phi} & \mathrm{Im}(\Phi) \subset \mathbb{C}U(\mathcal{H}_{n,k}) \\
& Tr_n \searrow & \downarrow \widetilde{Tr} \\
& & \mathbb{C}
\end{array}
$$

For this construction, we need the following lemma:

LEMMA 8.1. *The representation $\Phi : TL_n(d) \longrightarrow \mathbb{C}U(\mathcal{H}_{n,k})$ maps each ket $|p\rangle$ to a linear combination of kets each labeled by a path of the same length as the path p, and each having the same endpoint as p.*

An immediate corollary is:

COROLLARY 8.2. *Let $\mathcal{P}_{n,k,m}$ be the subset of $\mathcal{P}_{n,k}$ of all paths p in G_k of length n that start at the vertex 1 and end at the vertex m, where $1 \leq m < k$, and let $\mathcal{H}_{n,k,m}$ be the Hilbert subspace of $\mathcal{H}_{n,k}$ defined by the orthonormal basis $\{|p\rangle : p \in \mathcal{P}_{n,k,m}\}$. Then the representation $\Phi : TL_n(d) \longrightarrow \mathbb{C}U(\mathcal{H}_{n,k})$ splits into the direct sum of representations*

$$
\Phi = \bigoplus_{m=1}^{k-1} \Phi^{(m)} \quad ,
$$

where $\Phi^{(m)} : TL_n(d) \longrightarrow \mathbb{C}U(\mathcal{H}_{n,k,m})$ is the representation arising from the projection $\bigoplus_{m=1}^{k-1} \mathbb{C}U(\mathcal{H}_{n,k,m}) \longrightarrow \mathbb{C}U(\mathcal{H}_{n,k,m})$. Hence, the image $\mathrm{Im}(\Phi)$ of the representation $\Phi : TL_n(d) \longrightarrow \mathbb{C}U(\mathcal{H}_{n,k})$ lies in the direct sum of the algebras $\mathbb{C}U(\mathcal{H}_{n,k,m})$, $1 \leq m < k$, i.e.,

$$
\mathrm{Im}(\Phi) \subseteq \bigoplus_{m=1}^{k-1} \mathbb{C}U(\mathcal{H}_{n,k,m}) \quad .
$$

The above corollary gives us the latitude of searching for a compatible trace from among all the traces $\widetilde{Tr} : \mathrm{Im}(\Phi) \longrightarrow \mathbb{C}$ which are constructed by taking any linear combination of the **standard traces** $\widetilde{Tr}_m : \mathbb{C}U(\mathcal{H}_{n,k,m}) \longrightarrow \mathbb{C}$. Of these traces, the desired compatible trace is found to be the one given in the following theorem:

THEOREM 8.3. *Let $\lambda_\ell = \sin(\pi\ell/k)$ be the components of the eigenvector $\overrightarrow{\lambda}$ given in section 6, and let $N = \sum_{\ell=1}^{k-1} \lambda_\ell \dim(\mathcal{H}_{n,k,\ell})$. Then the trace $\widetilde{Tr} : \mathrm{Im}(\Phi) \longrightarrow \mathbb{C}$ defined by*

$$
\widetilde{Tr} = \frac{1}{N} \sum_{\ell=1}^{k-1} \lambda_\ell \widetilde{Tr}_\ell \quad ,
$$

is compatible with the Markov trace, i.e., $\widetilde{Tr}$ is a trace such that the diagram

$$
\begin{array}{ccc}
TL_n(d) & \xrightarrow{\Phi} & \mathrm{Im}(\Phi) \subseteq \bigoplus_{\ell=1}^{k-1} \mathbb{C}U(\mathcal{H}_{n,k,\ell}) \subset \mathbb{C}U(\mathcal{H}_{n,k}) \\
& Tr_n \searrow & \downarrow \widetilde{Tr} \\
& & \mathbb{C}
\end{array}
$$

is commutative. In other words, $\widetilde{Tr}$ is a trace such that $Tr_n = \widetilde{Tr} \circ \Phi$.

Since the Markov trace $Tr_n : TL_n(d) \longrightarrow \mathbb{C}$ is the unique trace satisfying the following three conditions

- $Tr_n(1) = 1$

- $Tr_n(XY) = Tr_n(YX)$ for all X and Y in $TL_n(d)$

- If $X \in TL_n(d)$, then $Tr_{n+1}(XE_n) = \frac{1}{d}Tr_n(X)$

the proof of the above theorem consists simply in verifying that $\widetilde{Tr} \circ \Phi : TL_n(d) \longrightarrow \mathbb{C}$ satisfies each of these conditions.

9. INTERMEDIATE SUMMARY

But where are we in regard to our objective of creating a quantum algorithm for approximating the value of the Jones polynomial at a root of unity of the form $e^{2\pi i/k}$, where k is an arbitrary positive integer?

For a knot (or link) given by the closure β^{Tr} of an n-stranded braid β, we have seen that the Jones polynomial is given by the expression
$$V_{\beta^{Tr}}(t) = -A^{2Writhe(\beta)}d^{n-1}Tr_n(\rho_A(\beta)) \ ,$$
where d and A are indeterminate complex numbers related by the equation $d = -A^2 - A^{-2}$, and where $t = A^{-4}$.

Setting $A = e^{-2\pi i/2k}$ (which implies $d = 2\cos(\pi/k)$ and $t = e^{2\pi i/k}$), we have the value of the Jones polynomial at $t = e^{2\pi i/k}$ is given by
$$V_{\beta^{Tr}}\left(e^{2\pi i/k}\right) = -A^{2Writhe(\beta)}d^{n-1}Tr_n(\rho_A(\beta)) \ .$$
Since $-A^{2Writhe(\beta)}d^{n-1}$ is easily computed, the task of determining $V_{\beta^{Tr}}\left(e^{2\pi i/k}\right)$ reduces to that of evaluating the trace
$$Tr_n(\rho_A(\beta)) \ .$$

But from the previous two sections, we have
$$Tr_n(\rho_A(\beta)) = \widetilde{Tr}_n[(\Phi \circ \rho_A)(\beta)] = \widetilde{Tr}_n\left[\left(\bigoplus_{m=1}^{k-1}\Phi^{(m)} \circ \rho_A\right)(\beta)\right] = \frac{1}{N}\sum_{m=1}^{k-1}\lambda_m Tr\left[\left(\Phi^{(m)} \circ \rho_A\right)(\beta)\right] \ ,$$

where Tr denotes the standard trace, where $\lambda_m = \sin(\pi m/k)$, and where $N = \sum_{m=1}^{k-1}\lambda_m \dim(\mathcal{H}_{n,k,m})$. Thus, our objective reduces to finding the trace of each of the following $k-1$ unitary transformations (called **global gates**)
$$U^{(m)} = \left(\Phi^{(m)} \circ \rho_A\right)(\beta), \quad 1 \leq m < k$$

If the knot (or link) is given by the closure of a braid β defined by the word
$$\beta = \prod_{\ell=1}^{L}b_{j(\ell)}^{\epsilon(\ell)} = b_{j(1)}^{\epsilon(1)}b_{j(2)}^{\epsilon(2)}\cdots b_{j(L)}^{\epsilon(L)} \ ,$$

where $b_1, b_2, \ldots, b_{n-1}$ are the generators of the braid group B_n, and where $\epsilon(i) = \pm 1$ for $i = 1, 2, \ldots, L$, then each unitary transformation $U^{(m)} = \left(\Phi^{(m)} \circ \rho_A \right)(\beta)$ is given by

$$U^{(m)} = \prod_{\ell=1}^{L} \left(U_{j(\ell)}^{(m)} \right)^{\epsilon(\ell)} \quad ,$$

where $U_j^{(m)}$ denotes the unitary transformation (called an **intermediate gate**)

$$U_j^{(m)} = \left(\Phi^{(m)} \circ \rho_A \right)(b_j) \quad , \quad 1 \le m < k, \ 1 \le j < n.$$

Thus, the trace $Tr_n \left(\rho_A \left(\beta \right) \right)$ we seek to approximate is given by the following expression

$$Tr_n \left(\rho_A \left(\beta \right) \right) = \frac{1}{N} \sum_{m=1}^{k-1} \lambda_m Tr \left[U^{(m)} \right] = \frac{1}{N} \sum_{m=1}^{k-1} \lambda_m Tr \left[\prod_{\ell=1}^{L} \left(U_{j(\ell)}^{(m)} \right)^{\epsilon(\ell)} \right] \quad .$$

We are finally in a position to describe the quantum algorithm found in[1] for approximating the Jones polynomial $V_{\beta^{Tr}}(t)$ at $t = e^{2\pi i/k}$ as a quantum algorithm consisting of the completion of two sequences of steps, called **phases**. The first is a preliminary phase called the **compilation phase**. After completion, the compilation phase is immediately followed by a second phase, called the **execution phase**.

10. THE COMPILATION PHASE
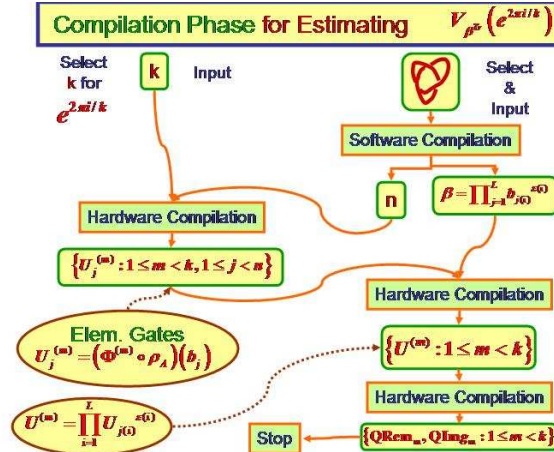
The compilation phase is illustrated in figure 25.

Figure 25. The compilation phase.

It consists of the following steps:

Software Compilation. On receiving a regular diagram of a knot (or link) K as input, the algorithm described by Alexander's theorem is executed to produce a regular diagram of a braid β whose closure gives the knot (or link) K. (See Birman.[4]) An algorithm, called **braid combing**, is then applied to the planar diagram of the braid β, producing as output a word

$$\beta = \prod_{\ell=1}^{L} b_{j(\ell)}^{\epsilon(\ell)} = b_{j(1)}^{\epsilon(1)} b_{j(2)}^{\epsilon(2)} \cdots b_{j(L)}^{\epsilon(L)}$$

describing the braid, and also as a side effect, producing as output the integer n giving the number of strands in β. (Once again, see Birman.[4]) *This word can be thought of as a computer program which will later be compiled into hardware.*

First Hardware Compilation. Upon receiving as input the integers k and n, use the Kitaev-Solovay,[281] theorem to implement (translate into hardware) good approximations of each intermediate gate $U_j^{(m)}$, $1 \le j < n$, $1 \le m < k$ as a product of polynomially many elementary gates.

Second Hardware Compilation. Use the braid word

$$\beta = \prod_{\ell=1}^{L} b_{j(\ell)}^{\epsilon(\ell)}$$

to implement (i.e., to physically construct) the global gates $U^{(m)}$ $1 \le m < k$ from the intermediate gates $U_j^{(m)}$. In other words, construct $U^{(m)}$ using the formula

$$U^{(m)} = \prod_{\ell=1}^{L} \left(U_{j(\ell)}^{(m)} \right)^{\epsilon(\ell)} \quad .$$

Third Hardware Compilation. For each m ($1 \le m < k$), construct from the global gate $U^{(m)}$ two quantum subroutines QRe_m and QIm_m, which upon input $|p\rangle$ ($p \in \mathcal{P}_{n,k,m}$), output at random a '0' or a '1' according to the following probability distributions

$$QRe_m\left(|p\rangle\right) = \begin{cases} 0 & \text{with probability } Prob\,(0) = \frac{1}{2} + \frac{1}{2}\operatorname{Re}\left\langle p|U^{(m)}|p\right\rangle \\ \\ 1 & \text{with probability } Prob\,(1) = \frac{1}{2} - \frac{1}{2}\operatorname{Re}\left\langle p|U^{(m)}|p\right\rangle \end{cases}$$

$$QIm_m\left(|p\rangle\right) = \begin{cases} 0 & \text{with probability } Prob\,(0) = \frac{1}{2} - \frac{1}{2}\operatorname{Im}\left\langle p|U^{(m)}|p\right\rangle \\ \\ 1 & \text{with probability } Prob\,(1) = \frac{1}{2} + \frac{1}{2}\operatorname{Im}\left\langle p|U^{(m)}|p\right\rangle \end{cases}$$

where $\operatorname{Re}\left\langle p|U^{(m)}|p\right\rangle$ and $\operatorname{Im}\left\langle p|U^{(m)}|p\right\rangle$ denote respectively the real and the imaginary parts of the bracket $\left\langle p|U^{(m)}|p\right\rangle$. Thus, if $QRe_m\left(|p\rangle\right)$ and $QIm_m\left(|p\rangle\right)$ are repeated executed, then we obtain respectively approximations of the real and imaginary parts of the bracket $\left\langle p|U^{(m)}|p\right\rangle$, as displayed below:

$$\begin{cases} (\#0's - \#1's)\,/\,(\#0's + \#1's) \approx \operatorname{Re}\left\langle p|U^{(m)}|p\right\rangle & \text{for } QRe_m\left(|p\rangle\right) \\ \\ (\#1's - \#0's)\,/\,(\#0's + \#1's) \approx \operatorname{Im}\left\langle p|U^{(m)}|p\right\rangle & \text{for } QIm_m\left(|p\rangle\right) \end{cases}$$

These two quantum subroutines QRe_m and QIm_m can be implemented using what has now come to be known as the **Hadamard test**. A wiring diagram defining the quantum subroutine $QRe_m\left(|p\rangle\right)$ is given in figure 26. Input to QRe_m consists of the state $|p\rangle$ and an ancillary qubit in state $|0\rangle$. After execution of the the wiring diagram, measurement of the ancillary qubit with respect to the standard basis will produce the desired probability distribution. Similarly, Figure 27 defines the quantum subroutine QIm_m.

$$\text{Prob}(0) = \frac{1}{2} + \frac{1}{2}\text{Re}\langle p|U|p\rangle$$

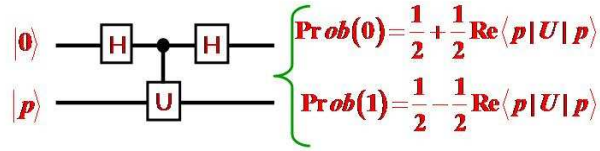$$\text{Prob}(1) = \frac{1}{2} - \frac{1}{2}\text{Re}\langle p|U|p\rangle$$

Figure 26. A wiring diagram describing the quantum subroutine QRe_m. Input consists of the state $|p\rangle$ and an ancillarary qubit in state $|0\rangle$. The Hadamard gate is denoted by H. After execution of the first hadard gate, the ancillary qubit is used to control the global gate $U^{(m)}$. Measurement of the ancillary qubit after the execution of the second Hadamard gate produces the desired probability distribution.

$$\text{Prob}(0) = \frac{1}{2} - \frac{1}{2}\text{Im}\langle p|U|p\rangle$$

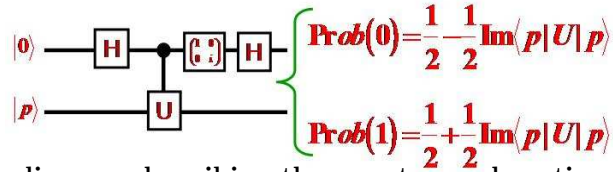$$\text{Prob}(1) = \frac{1}{2} + \frac{1}{2}\text{Im}\langle p|U|p\rangle$$

Figure 27. A wiring diagram describing the quantum subroutine QIm_m. This subroutine is the same as the quantum subroutine QRe_m except for the additional single qubit phase gate
$$\begin{pmatrix} 1 & 0 \\ 0 & i \end{pmatrix}.$$

11. THE EXECUTION PHASE

The compilation phase is then followed by the execution phase as described by the pseudocode given below:

Execution Phase for Estimating $\mathbf{V}_{\beta^{Tr}}\left(e^{2\pi i/k}\right)$

$\textsc{Algorithm}\ AJK\ (n,k)$
  $Trace = 0$
  $\textsc{loop}\ \ m = 1 \ldots k-1$
    $Trace_m = 0\ \ \textsc{and}\ \ \ \lambda_m = \sin\left(\pi m/k\right)$
    $\textsc{loop}\ \ p \in \mathcal{P}_{n,k,m}$
      $Re = 0\ \ \textsc{and}\ \ Im = 0$
      $\textsc{loop}\ \ Iteration = 1 \ldots Number of Iterations$
        $RealBit = QRe_m\left(p\right)\ \ \textsc{and}\ \ \ ImgBit = QIm_m\left(p\right)$
        $Re = Re + \left(-1\right)^{RealBit}\ \ \ \textsc{and}\ \ \ \ Im = Im - \left(-1\right)^{ImgBit}$
      $\textsc{end}\ \ Iteration\text{-}\textsc{loop}$
      $Trace_m = Trace_m + \left(Re + i * Im\right)/Number of Iterations$
    $\textsc{end}\ p\text{-}\textsc{loop}$
    $Trace = Trace + \lambda_m * Trace_m$
  $\textsc{end}\ m\text{-}\textsc{loop}$
  $\textsc{output}\ Trace$
$\textsc{end}\ \textsc{Algorithm}\ AJK$

This phase consists of three nested loops. The innermost loop calls the quantum subroutines QRe_m and QIm_m. The parameter $Number of Iterations$ is chosen according to the Chernoof-Hoeffding bound to provide the desired accuracy for the approximation. If resources are available, the outermost iteration loop can be replaced by a parallel implementation.

12. CONCLUSION

Indeed, much more could be said about the AJL quantum algorithm for the Jones polynomial. But, for the time being, we will leave that task to one of our future forthcoming papers on this subject.

13. ACKNOWLEDGEMENTS

REFERENCES

1. Aharonov, Dorit, Vaughan Jones, and Zeph Landau, **A polynomial quantum algorithm for approximating the Jones polynomial**, http://arxiv.org/abs/quant-ph/0511096
2. Artin, E., **Theory of braids**, Annals of Mathematics, **48**, (1947), 101-126.
3. Bernstein, E., and U. Vazirani, **Quantum complexity theory**, SIAM Journal of Computation, 26, 5, (1997), 1411-1473.
4. Birman, J., **"Braids, links and mapping class groups,"** Annals of Mathematical Studies, vol. 82, (1974).

5. Crowell, Richard H., and Ralph H. Fox, **"Inroduction to Knot Theory,"** Springer-Verlag, (1977).

6. Ewing, B. and K.C. Millett, **A load balanced algorithm for the calculation of the polynomial knot and link invariants** in **"The mathematical heritage of C. F. Gauss,"** 225–266, World Sci. Publishing, River Edge, NJ, (1991).

7. Fox, R.H., **A quick trip through knot theory**, in **"Topology of 3-Manifolds and Related Topics,"** M.K. Fort, Jr., (Ed.), Printice-Hall, (1962).

8. Freedman, M., **P/NP and the quantum field computer**, Proc. Natl. Acad. Sci., USA, 95, (1998), 98–101

9. Freedman, M., A.Kitaev, M. Larsen, and Z. Wang, **Topological quantum computation** in **"Mathematical challenges of the 21st century (Los Angeles, CA, 2000),"** Bull. Amer. Math. Soc., 40 , no. 1, (2003), 31–38

10. Freedman, M.H., A. Kitaev, and Z. Wang, **Simulation of topological field theories by quantum computers**, Commun. Math. Phys., 227, (2002) 587-603.

11. Freedman, M.H., M. Larsen, and Z. Wang, A modular Functor which is universal for quantum computation, Commun. Math. Phys., 227 no. 3, (2002), 605-622.

12. Goodman, F.M., P. de la Harpe, and V.F.R. Jones, **Coxeter graphs and towers of algebras**, Springer-Verlag, (1989).

13. Garnerone, Silvano, Annalisa Marzuoli, and Mario Rasetti, **Quantum automata, braid group and link polynomials**, http://arxiv.org/abs/quant-ph/0601169

14. Jaeger, F., D.L. Vertigan, D.J.A. Welsh, **On the computational complexity of the Jones and Tutte polynomials**, Math. Proc. Cambridge Philos. Soc., 108, no. 1, (1990), 35–53.

15. Jones, V.F.R., **A polynomial invariant for knots via von Neumann algebras**, Bull. Amer. Math. Soc., 12, no. 1, (1985), 103–111.

16. Jones, V.F.R., **Index for subfactors**, Invent. Math 72 (1983), 1–25.

17. Jones, V.F.R., **Braid groups, Hecke Algebras and type II factors**, in **"Geometric methods in Operator Algebras,"** Pitman Research Notes in Math., 123 (1986), 242–273.

18. Kauffman, L., **State models and the Jones polynomial**, Topology 26, (1987),395-407.

19. Kauffman, Louis H., and Sostenes L. Lins, **"Temperley-Lieb Recoupling Theory and Invariants of 3-Manifolds,"** Princeton University Press, (1994).

20. Kauffman, Louis H., **"Knots and Physics,"** (second edition), World Scientific, (1993)

21. Kauffman, Louis H., **Quantum computing and the Jones polynomial** in **"Quantum computation and information,"** edited by S.J. Lomonaco, Jr., AMS CONM/305, (2000), 101-137.

22. Kuperberg, G., **A subexponential-time quantum algorithm for the dihedral hidden subgroup problem**, http://arxiv.org/abs/quant-ph/0302112 .

23. Lickorish, W.B. Raymond, **"An Introduction to Knot Theory,"** Springer-Verlag, (1997).

24. Lomonaco, Samuel J., Jr., **"Quantum Computation: A Grand Mathematical Challenge for the Twenty-First Century and the Millennium,"** AMS PSAPM 58, (2000).

25. Lomonaco, Samuel J., Jr., (ed.), **"Low Dimensional Topology,"** AMS CONM/20, (1981).

26. Markov, A., **Uber de freie Aquivalenz geschlossener Zopfe**, Rossiiskaya Akademiya Nauk, Matematicheskii Sbornik, 1, (1935), 73–78.

27. Murasugi, Kunio, **"Knot Theory and Its Applications,"** Birkhauser, (1996).

28. Neilsen, Michael A., and Isaac L. Chuang, **"Quantum Computation and Quantum Information,"** Cambridge University Press, (2000).

29. Roldsen, Dale, **"Knots and Links,"** Publish or Perish, Inc., (1976).

30. Vogel, P., **Representation of links by braids: A new algorithm,** Comment. math. Helvetici, 65 (1) (1990), 104–113.

31. Welsh, D.J.A., **The Computational Complexity of Some Classical Problems from Statistical Physics**, in **"Disorder in Physical Systems,"** G.R. Grimmett and D.J.A. Welsh, eds., Clarendon Press, Oxford, (1990), 307-321.

32. Welsh, D.J.A., "Complexity: Knots, Colourings and Counting," Cambridge University Press, (2000).

33. Witten, Edward, **Quantum field theory and the Jones polynomial**, Comm. Math. Phys., 121, no. 3, (1989), 351–399.

34. Wu, F.Y., **Knot Theory and statistical mechanics**, Rev. Mod. Phys. 64, No. 4., (1992).
35. Wocjan, Pawel, and Jon Yard, **The Jones polynomial: quantum algorithms and applications in quantum complexity theory**, http://arxiv.org/abs/quant-ph/0603069